

FIG.1

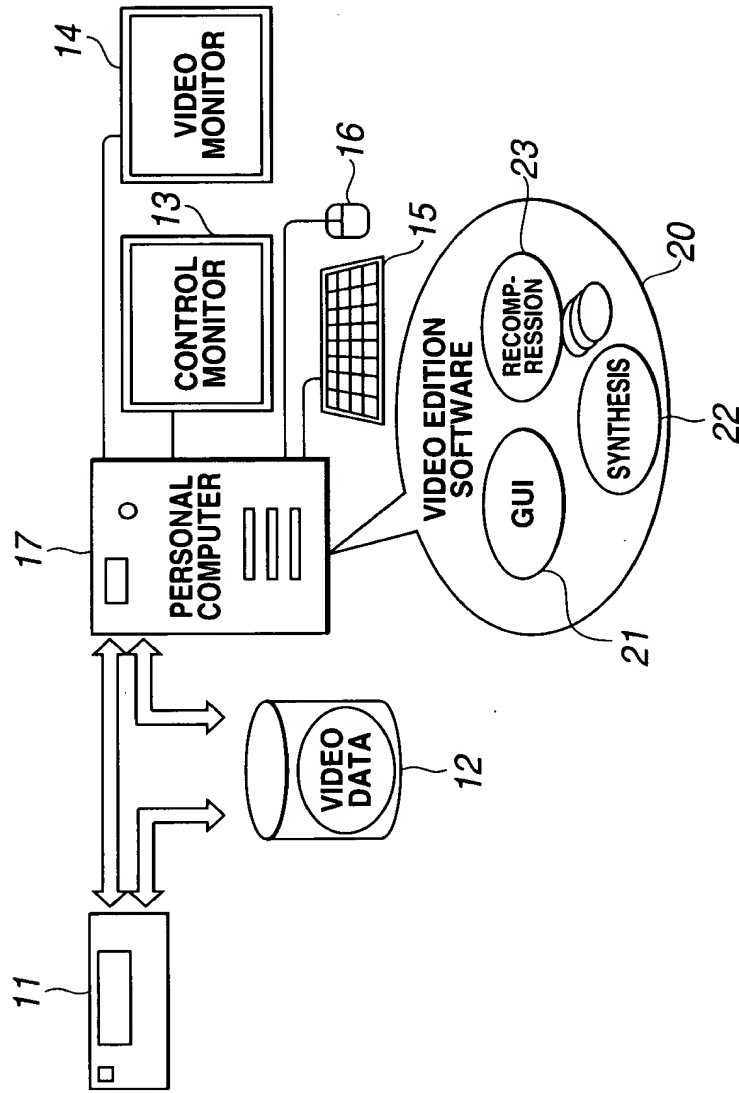
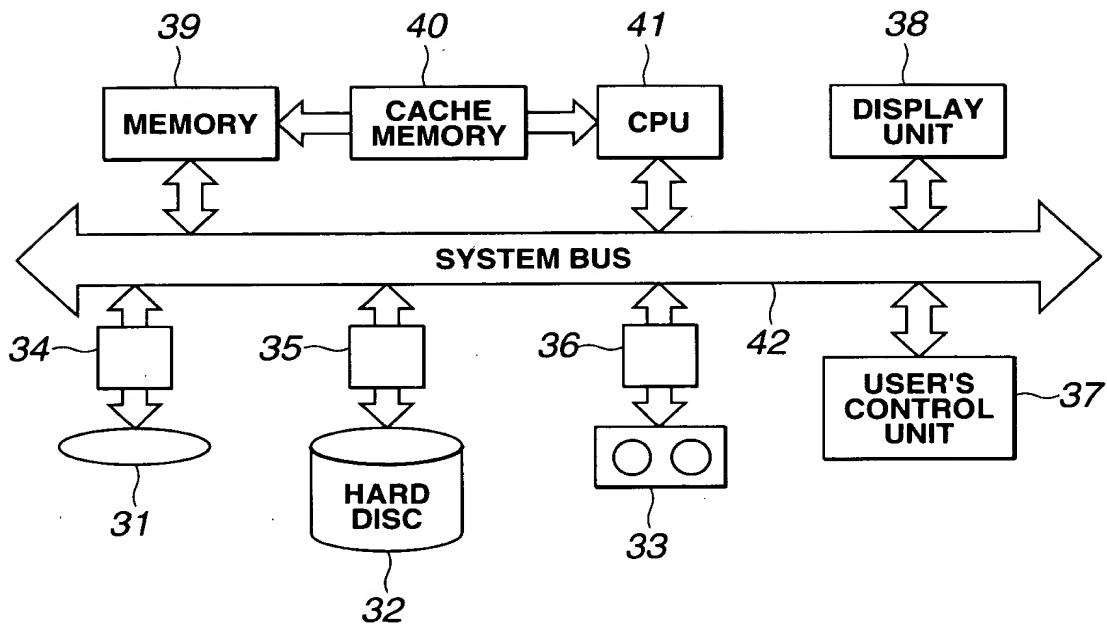


FIG.2

**FIG.3**

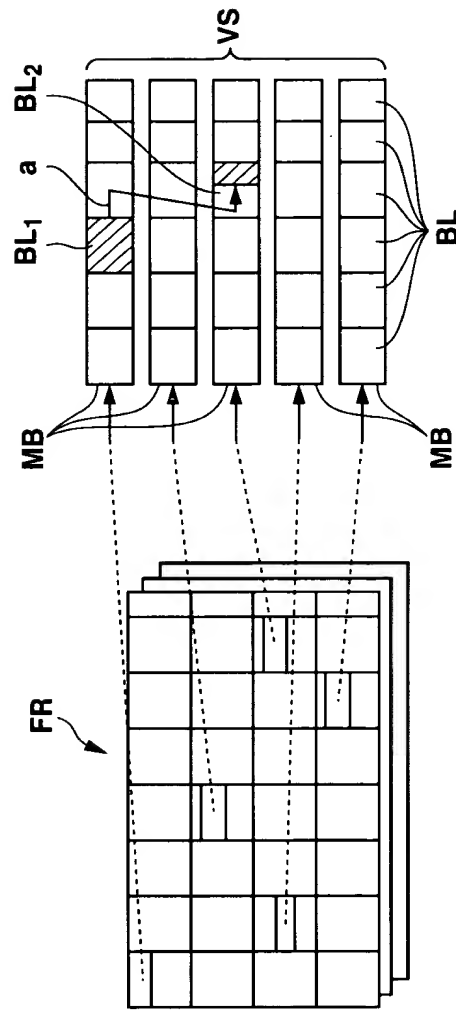


FIG.4

5/25

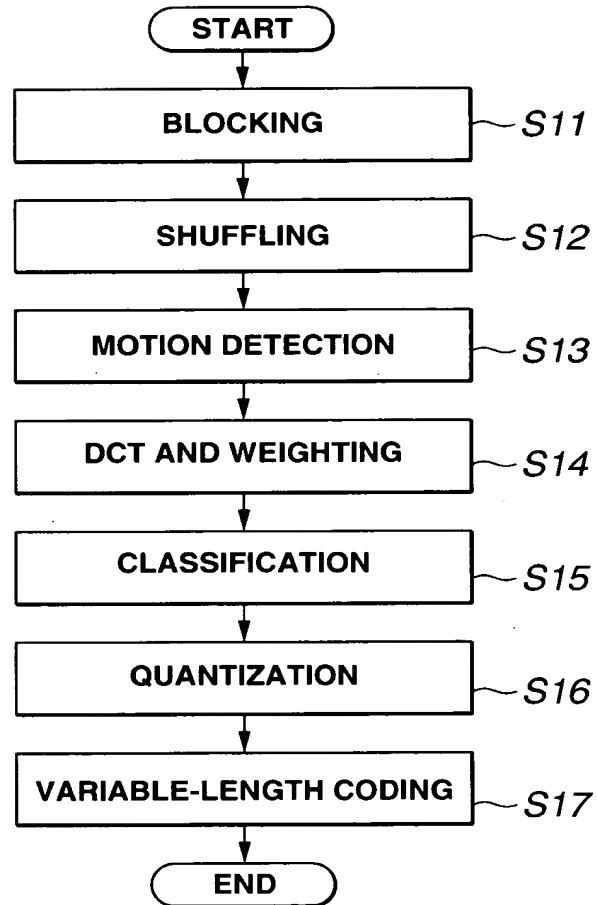


FIG.5

6/25

	CLASS NO.(CNO)				AREA NO.(ANO)			
	0	1	2	3	0	1	2	3
QUANTIZER NO.(QNO)	15				1	1	1	1
	14				1	1	1	1
	13				1	1	1	1
	12	15			1	1	1	1
	11	14			1	1	1	1
	10	13		15	1	1	1	1
	9	12	15	14	1	1	1	1
	8	11	14	13	1	1	1	2
	7	10	13	12	1	1	2	2
	6	9	12	11	1	1	2	2
	5	8	11	10	1	2	2	4
	4	7	10	9	1	2	2	4
	3	6	9	8	2	2	4	4
	2	5	8	7	2	2	4	4
	1	4	7	6	2	4	4	8
	0	3	6	5	2	4	4	8
		2	5	4	4	4	8	8
		1	4	3	4	4	8	8
		0	3	2	4	8	8	16
			2	1	4	8	8	16
			1	0	8	8	16	16
			0		8	8	16	16

FIG.6

T050E0" 06598260

7/25

HORIZONTAL
→

	0	1	2	3	4	5	6	7
0	DC	0	0	1	1	1	2	2
1	0	0	1	1	1	2	2	2
2	0	1	1	1	2	2	2	3
3	1	1	1	2	2	2	3	3
4	1	1	2	2	2	3	3	3
5	1	2	2	2	3	3	3	3
6	2	2	2	3	3	3	3	3
7	2	2	3	3	3	3	3	3

↓
VERTICAL

FIG.7A

HORIZONTAL
→

	0	1	2	3	4	5	6	7
0	DC	0	1	1	1	2	2	3
1	0	1	1	2	2	2	3	3
2	1	1	2	2	2	3	3	3
3	1	2	2	2	3	3	3	3
4	0	0	1	1	2	2	2	3
5	0	1	1	2	2	2	3	3
6	1	1	2	2	2	3	3	3
7	1	2	2	3	3	3	3	3

(SUM)
↓
VERTICAL
(DIFFERENCE)

FIG.7B

09/786590-06598/60

8/25

HORIZONTAL
→

	0	1	2	3	4	5	6	7
0	1	2	6	7	15	16	28	29
1	3	5	8	14	17	27	30	43
2	4	9	13	18	26	31	42	44
3	10	12	19	25	32	41	45	54
4	11	20	24	33	40	46	53	55
5	21	23	34	39	47	52	56	61
6	22	35	38	48	51	57	60	62
7	36	37	49	50	58	59	63	64

VERTICAL
↓

FIG.8A

HORIZONTAL
→

	0	1	2	3	4	5	6	7
0	1	3	7	19	21	35	37	51
1	5	9	17	23	33	39	49	53
2	11	15	25	31	41	47	55	61
3	13	27	29	43	45	57	59	63
4	2	4	8	20	22	36	38	52
5	6	10	18	24	34	40	50	54
6	12	16	26	32	42	48	56	62
7	14	28	30	44	46	58	60	64

(SUM)
↓
VERTICAL
(DIFFERENCE)

FIG.8B

09/786590-03001

9/25

(run, amp)		CODE	LENGTH	(run, amp)		CODE	LENGTH
0	1	00s	2+1	11	1	111100000s	9+1
0	2	010s	3+1	12	1	111100001s	
EOB		0110	4	13	1	111100010s	
1	1	0111s	4+1	14	1	111100011s	
0	3	1000s		5	2	111100100s	
0	4	1001s		6	2	111100101s	
2	1	10100s	5+1	3	3	111100110s	
1	2	10101s		4	3	111100111s	
0	5	10110s		2	4	111101000s	
0	6	10111s		2	5	111101001s	
3	1	110000s	6+1	1	8	111101010s	
4	1	110001s		0	18	111101011s	
0	7	110010s		0	19	111101100s	
0	8	110011s		0	20	111101101s	
5	1	1101000s	7+1	0	21	111101110s	
6	1	1101001s		0	22	111101111s	
2	2	1101010s		5	3	1111100000s	10+1
1	3	1101011s		3	4	1111100001s	
1	4	1101100s		3	5	1111100010s	
0	9	1101101s		2	6	1111100011s	
0	10	1101110s		1	9	1111100100s	
0	11	1101111s		1	10	1111100101s	
7	1	11100000s	8+1	1	11	1111100110s	11
8	1	11100001s		0	0	11111001110s	
9	1	11100010s		1	0	11111001111s	
10	1	11100011s		6	3	11111010000s	11+1
3	2	11100100s		4	4	11111010001s	
4	2	11100101s		3	6	11111010010s	
2	3	11100110s		1	12	11111010011s	
1	5	11100111s		1	13	11111010100s	
1	6	11101000s		1	14	11111010101s	
1	7	11101001s		2	0	111110101100	12
0	12	11101010s		3	0	111110101101	
0	13	11101011s		4	0	111110101110	
0	14	11101100s		5	0	111110101111	
0	15	11101101s					
0	16	11101110s					
0	17	11101111s					

FIG.9

T050E0-06598/60

10/25

(run, amp)		CODE		LENGTH	
7	2	111110110000s		12+1	
8	2	111110110001s			
9	2	111110110010s			
10	2	111110110011s			
7	3	111110110100s			
8	3	111110110101s			
4	5	111110110110s			
3	7	111110110111s			
2	7	111110111000s			
2	8	111110111001s			
2	9	111110111010s			
2	10	111110111011s			
2	11	111110111100s			
1	15	111110111101s			
1	16	111110111110s			
1	17	111110111111s			
6	0	1111110000110		13	
7	0	1111110000111			
⋮	⋮	1111110	BINARY NOTATION OF R: R=6 TO 61		
R	0				
⋮	⋮				
61	0	1111110111101			
0	23	111111100010111s		15+1	
0	24	111111111111000s			
⋮	⋮	1111111	BINARY NOTATION OF A: A=23 TO 255		s
0	A				
⋮	⋮				
0	255	111111111111111s			

FIG.10

050301

09/786590

11/25

[illegible]

FIG. 11

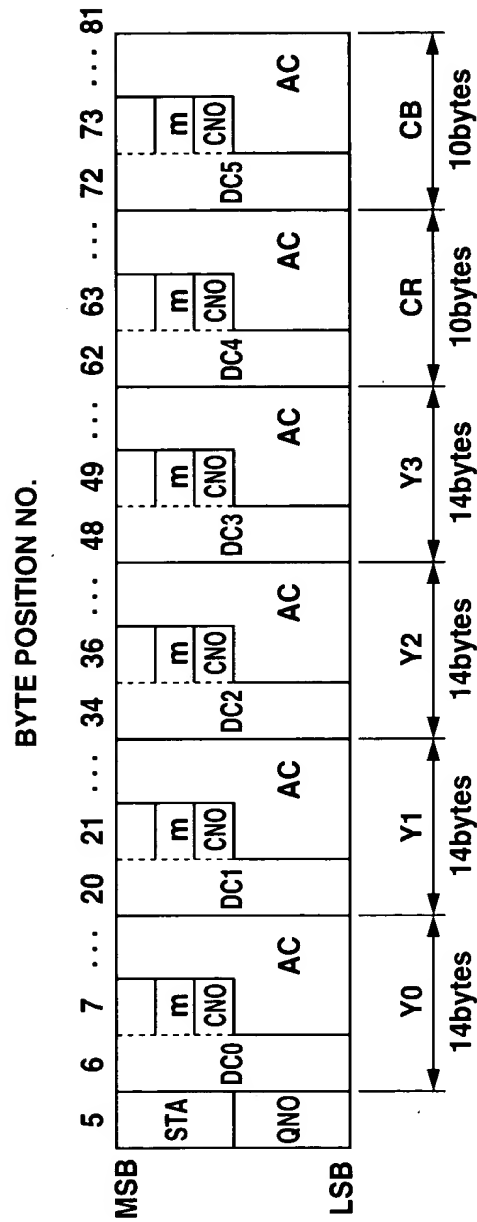


FIG.12



FIG. 13B

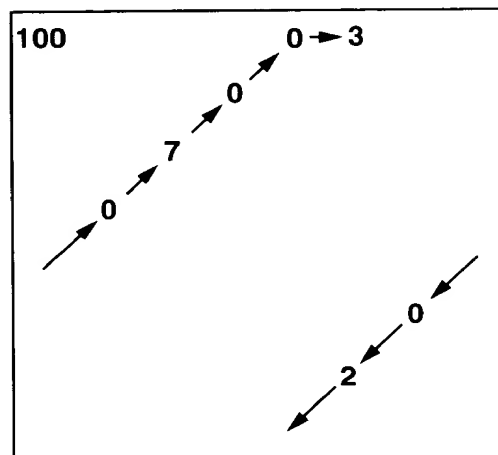


FIG.14A

... 0, 15, 0, 0, 7, 0, 9, ...

FIG.14B

1111101111010, 1111101110001, ... 11111001000

13 13 11

FIG.14C

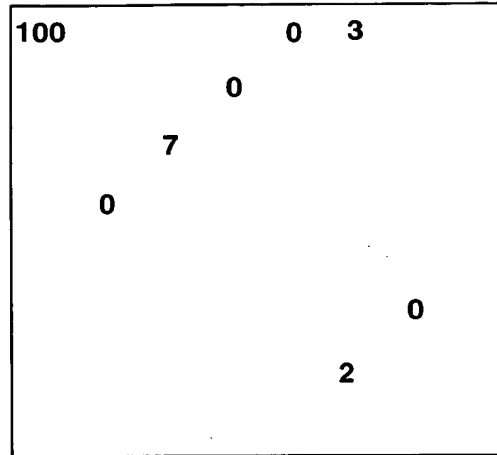
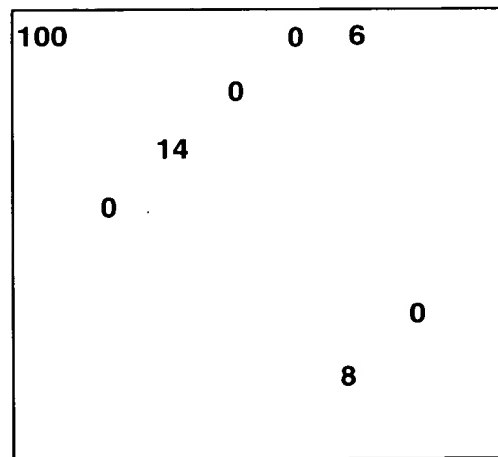
... 0, 7, 0, 0, 3, 0, 2, ...

FIG.14D

111010010, 111001100, ... 101010

9 9 6

15/25

**FIG.15A****FIG.15B**

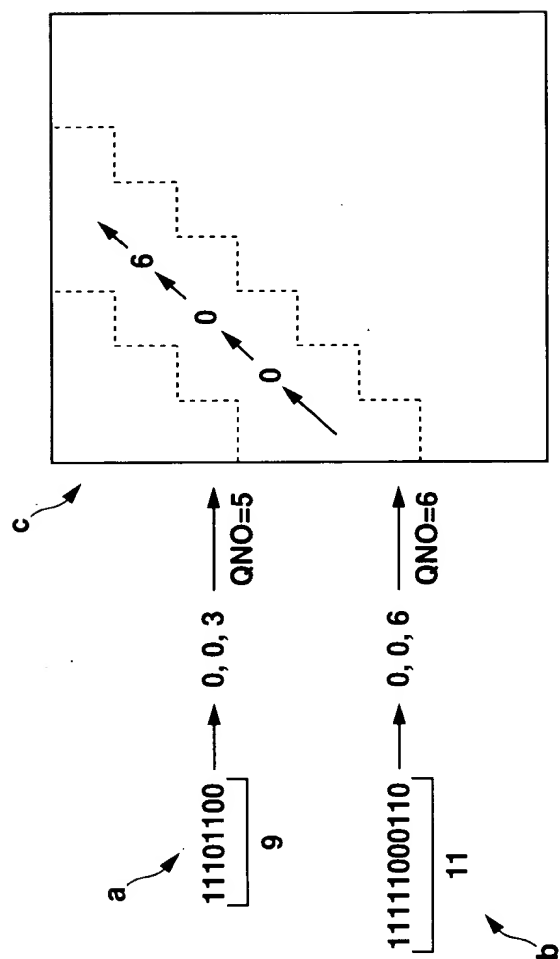


FIG.16

17/25

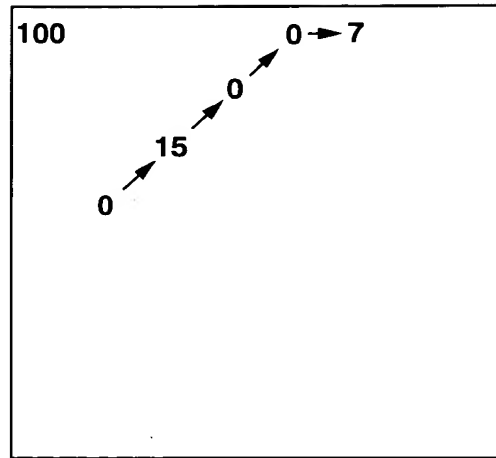
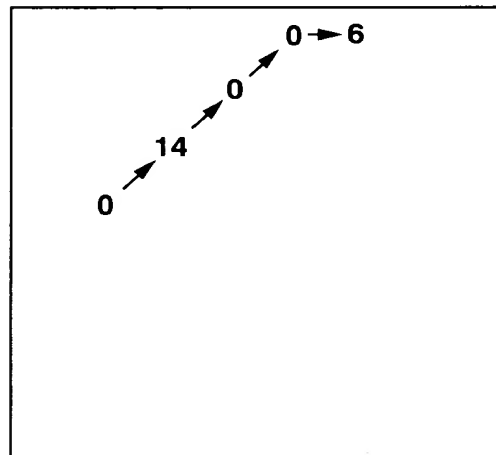
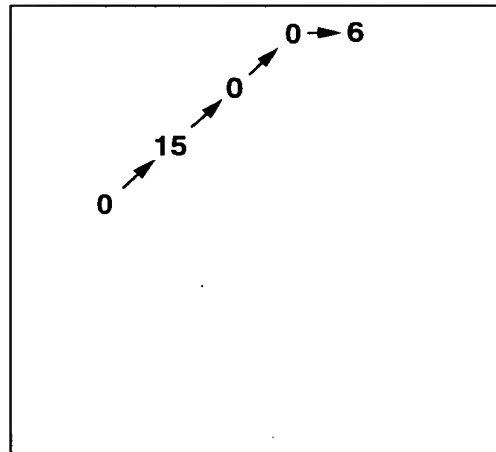
**FIG.17A****FIG.17B****FIG.17C**

FIG. 18A QNO=6 → 1111101111010, 0, 15 (1, 15)
1111101110001, 0, 17 (2, 7)

FIG. 18B QNO=5 → 111010010, 0, 7 (1, 7)
111001100, 0, 0, 3 (2, 3)

FIG. 18C QNO=6 → 1111101111010, 0, 15 (1, 15)
11111000110, 0, 0, 6 (2, 6)

```

static int sHuffmanReducible(int nAmp, int nRun)
{
    if(nAmp<0)
        nAmp=-nAmp;
    if(((nAmp<3)|| (nAmp>23))
        return 0;
    int nThisLength=sHuffmanLength(nAmp, nRun);
    int nNewLength=sHuffmanLength(nAmp-1, nRun);
    int nReduced=nThisLength-nNewLength;
    return(nReduced>0)?nReduced; 0;
}

static int sHuffmanReducibleToAmp0(int nAmp, int nRun, int nNextAmp, int nNextRun)
{
    int nThisLength=sHuffmanLength(nAmp, nRun);
    if(nNextRun==EOB){/*the end of data*/
        return nThisLength;
    }
    int nNextLength=sHuffmanLength(nNextAmp, nNextRun);
    int nNewLength=sHuffmanLength(nNextAmp, nRun+1+nNextRun);
    return nThisLength+nNextLength-nNewLength;
}

```

FIG.19

20/25

```

static BOOL sQStepReductionTable0[4][16] = {
    {0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0},
    {0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0}
};
static BOOL sQStepReductionTable1[4][16] = {
    {0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0},
    {0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0}
};
static BOOL sQStepReductionTable2[4][16] = {
    {0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0},
    {0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0},
    {0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0}
};
static BOOL sQStepReductionTable3[4][16] = {
    {0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0},
    {0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0},
    {0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1},
    {0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0}
};

static BOOL sQStepReducible(int nQno, int nClassNo, int nAreaNo)
{
    switch (nAreaNo) {
        case (3) :
            return sQStepReductionTable3[nClassNo][nQno];
        case (2) :
            return sQStepReductionTable2[nClassNo][nQno];
        case (1) :
            return sQStepReductionTable1[nClassNo][nQno];
        case (0) :
            return sQStepReductionTable0[nClassNo][nQno];
        default :
            return FALSE ;
    }
}

```

FIG.20

21/25

#define EOB 62

```
int RequantizeRunAmp(short* pRuns, short* pAmps, int nReducingSize, int nQno, int nClassNo)
{
```

```
    static int area[4] = {1, 6, 21, 43};

    int nAreaEnd[4];

    for (int i = 0; i < 4; i++) {
        nAreaEnd[i] = -1;
    }

    int nArea = 1;
    int nCoef = 0;

    for (i = 0; pRuns[i] != EOB; i++) {
        if (pRuns[i] > EOB) // skip invalid entry
            continue;

        nCoef += pRuns[i] + 1;

        if ((nArea < 4) && (nCoef >= area[nArea])) {
            nAreaEnd[nArea-1] = i-1;
            nArea++;
        }
    }

    int nLastEntry = i-1;

    for (i = 0; i < 4; i++) {
        if (nAreaEnd[i] == -1) {
            nAreaEnd[i] = nLastEntry;
        }
    }
```

```
int nTotalReducedSize = 0;
```

```
int nReducedSize;
int nLastAmp;
int nLastRun;
int nLastPos;
```

FIG.21

09/786590-030501

22/25

for (i = 3; i > 0; i--) {

if (sQStepReducible (nQno, nClassNo, i))

```

    if (nAreaEnd[i] < nLastEntry) {
        nLastPos = nAreaEnd[i]+1;
        nLastRun = pRuns[nLastPos];
        nLastAmp = pAmps[nLastPos];
    } else {
        nLastPos = nLastEntry;
        nLastRun = EOB;
    }

```

int nPrevAreaEnd = (i>0)?nAreaEnd[i-1]: -1;

/*in present area, entires are replaced one after another beginning with higher-order ones*/

for (int j = nAreaEnd[i]; j > nPrevAreaEnd; j--) {

```

    if (pRuns[j] >= EOB)
        continue;

```

/*when amp is 1 or -1 or 0*/

```

    if ((pAmps[j] <= 1) || (pAmps[j] >= -1)) {
        nReducedSize = sHuffmanReducibleToAmp0(pAmps[j], pRuns[j], nLastAmp, nLastRun);

```

```

        if (nReducedSize >= 0) {

```

```

            if (nLastRun == EOB) {

```

```

                pRuns[j] = EOB; // set the end of data

```

```

            } else {

```

```

                pAmps[j] = nLastAmp;

```

```

                pRuns[j] += nLastRun + 1; // run is add

```

```

                pRuns[nLastPos] = 255; // marking to indicate that entry is invalid

```

```

            }

```

```

        } else {

```

```

            /*amp takes other value*/

```

```

            nReducedSize = sHuffmanReducible(pAmps[j], pRuns[j]);

```

```

            if (nReducedSize > 0)

```

```

                pAmps[j] -= (pAmps[j] > 0) ? 1 : (-1); /*amp is replaced*/

```

```

        }

```

```

    if (nReducedSize > 0) {

```

```

        nTotalReducedSize += nReducedSize;

```

```

        if (nTotalReducedSize >= nReducingSize)

```

```

            return nTotalReducedSize; /*end since target reduction has attained*/

```

```

    }

```

```

    nLastAmp = pAmps[j];

```

```

    nLastRun = pRuns[j];

```

```

    nLastPos = j;

```

```

}

```

```

return nTotalReducedSize;

```

FIG.22

INPUT CODE WORD	NEW CODE WORD	INPUT(run, amp) →NEW(run, amp)
1000s	010s	(0, 3) → (0, 2)
10110s	1001s	(0, 5) → (0, 4)
110010s	1011s	(0, 7) → (0, 6)
1101101s	110011s	(0, 9) → (0, 8)
111111100010111s	111101111s	(0, 23) → (0, 22)
1101011s	10101s	(1, 3) → (1, 2)
11100111s	1101100s	(1, 5) → (1, 4)
1111100100s	111101010s	(1, 9) → (1, 8)
111110111101s	11111010101s	(1, 15) → (1, 14)
11100110s	1101010s	(2, 3) → (2, 2)
111110111000s	1111100011s	(2, 7) → (2, 6)
111100110s	11100100s	(3, 3) → (3, 2)
111110110111s	11111010010s	(3, 7) → (3, 6)
111100111s	11100101s	(4, 3) → (4, 2)
111110110110s	11111010001s	(4, 5) → (4, 4)
1111100000s	111100100s	(5, 3) → (5, 2)
11111010000s	111100101s	(6, 3) → (6, 2)

FIG.23

09/786590-030501

	Y ₀	Y ₁	Y ₂	Y ₃	Cr	Cb
MB0	B ₀	B ₀₁	B ₀₅
MB1	B ₁₀	B ₁₁				
MB2	B ₂₀	⋮	⋱			
MB3	B ₃₀	⋮		⋱		
MB4	B ₄₀	⋮				B ₄₅

FIG.24

25/25

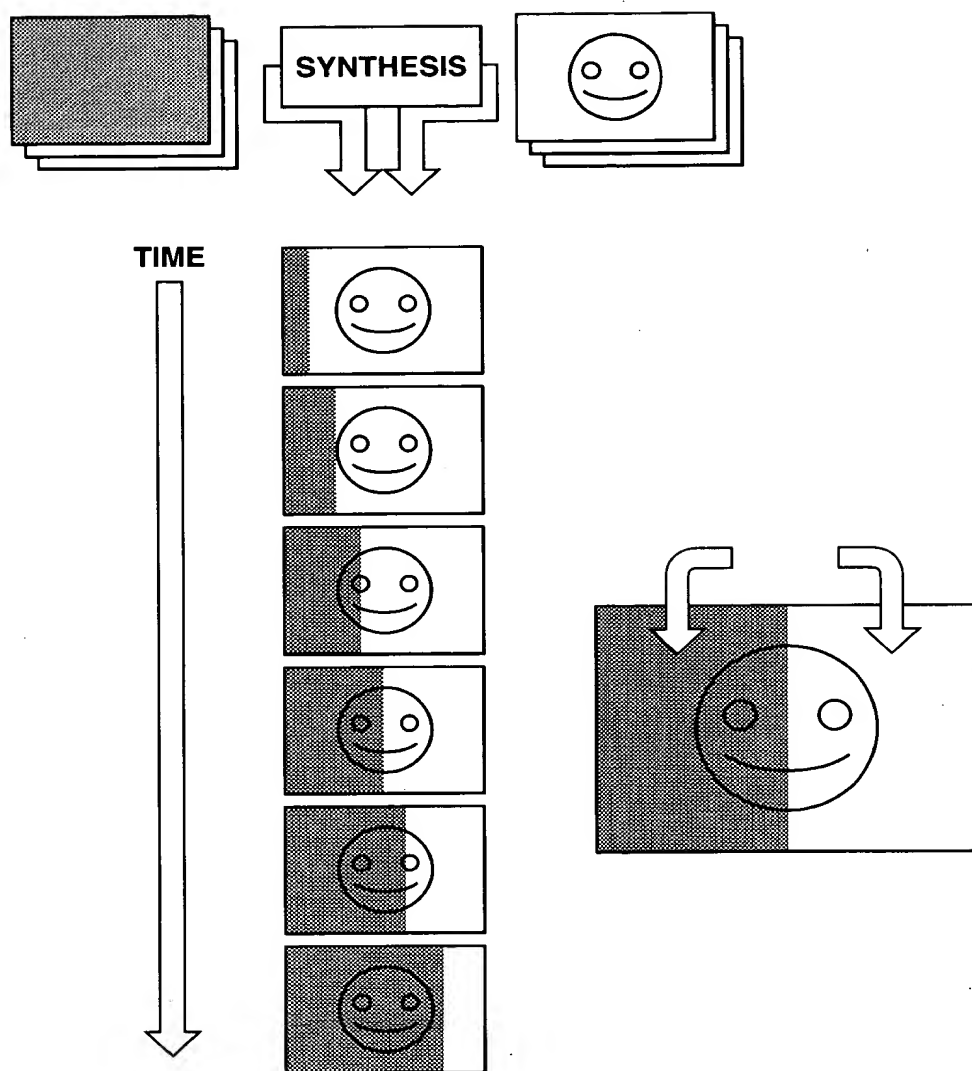


FIG.25